CLAIMS

What is Claimed is:

1. A method for arithmetic expression optimization, comprising:

receiving a first instruction defined for a first processor having a first base, said instruction including an operator and at least one operand;

converting said first instruction to a second instruction optimized for a second processor having a second base when said at least one operand does not carry potential overflow beyond said second base or when said operator is insensitive to overflow, said second base smaller than said first base; and

converting to a wider base a third instruction that is the source of potential overflow associated with said at least one operand when said at least one operand carries the potential for overflow beyond said second base and when said operator is sensitive to overflow, said third instruction having been previously optimized, said wider base larger than said second base and smaller or equal to said first base.

2. The method of claim 1 wherein said converting to a wider base further comprises discarding previous conversion results of said third instruction before said converting to a wider base.

3. The method of claim 1, further comprising rejecting an expression that cannot be optimized to a smaller base on said second processor.

4. The method of claim 1 wherein said converting to a wider base further comprises rejecting said first instruction when said wider base is not supported by said second processor.

5. The method of claim 1 wherein said first instruction is arithmetic.

6. The method of claim 1 wherein said first instruction comprises a non-arithmetic, type-sensitive instruction.

7. The method of claim 5, further comprising, after said converting said first instruction, returning to receiving said first instruction until all instructions defined for said first processor are converted.

8. The method of claim 7, further comprising linking each instruction to successor instructions in all control paths.

9. The method of claim 8 wherein said converting said first instruction further comprises:

linking each result of an instruction to all instructions that consume said result;

if said converting includes creating a value, linking said value to the instruction that produced said value; and

if said value carries a potential for overflow, linking said value to the instruction that originally caused said overflow.

10. The method of claim 1 wherein

said first processor comprises a Java™ Virtual Machine; and

said second processor comprises a Java Card™ Virtual Machine.

11. The method of claim 1 wherein

said first base is used by said first processor for performing arithmetic operations on at least one data type, said at least one data type having a size less than the size of said first base; and

said second base is used by said second processor for performing arithmetic operations on said at least one data type, said second base having a size equal to the size of said at least one data type.

12. The method of claim 1 wherein

said first processor comprises a 32-bit processor; and

said second processor comprises a resource-constrained 16-bit processor.

13. The method of claim 9 wherein

said first base is used by said first processor for performing arithmetic operations on at least one data type, said at least one data type having a size less than the size of said first base; and

said second base is used by said second processor for performing arithmetic operations on said at least one data type, said second base having a size greater than the size of said at least one data type.

14. The method of claim 13 wherein

said first processor comprises a 32-bit processor; and

said second processor comprises a resource-constrained 16-bit processor.

15. A program storage device readable by a machine, embodying a program of instructions executable by the machine to perform arithmetic expression optimization comprising:

receiving a first instruction defined for a first processor having a first base, said first instruction including an operator and at least one operand;

converting said first instruction to a second instruction optimized for a second processor having a second base when said at least one operand does not carry

potential overflow beyond said second base or when said operator is insensitive to overflow, said second base smaller than said first base; and

converting to a wider base a third instruction that is the source of potential overflow associated with said at least one operand when said at least one operand carries the potential for overflow beyond said second base and when said operator is sensitive to overflow, said third instruction having been previously optimized, said wider base larger than said second base and smaller or equal to said first base.

16. The program storage device of claim 15 wherein said converting to a wider base further comprises discarding previous conversion results of said third instruction before said converting to a wider base.

17. The program storage device of claim 15, further comprising rejecting an expression that cannot be optimized to a smaller base on said second processor.

18. The program storage device of claim 15 wherein said converting to a wider base further comprises rejecting said first instruction when said wider base is not supported by said second processor.

19. The program storage device of claim 15 wherein said first instruction is arithmetic.

20. The program storage device of claim 15 wherein said first instruction comprises a non-arithmetic, type-sensitive instruction.

21. The program storage device of claim 19, further comprising, after said converting said first instruction, returning to receiving said first instruction until all instructions defined for said first processor are converted.

22. The program storage device of claim 21, further comprising linking each instruction to successor instructions in all control paths.

23. The program storage device of claim 22 wherein said converting said first instruction further comprises:

linking each result of an instruction to all instructions that consume said result;

if said converting includes creating a value, linking said value to the instruction that produced said value; and

if said value carries a potential for overflow, linking said value to the instruction that originally caused said overflow.

24. The program storage device of claim 15 wherein

said first processor comprises a Java™ Virtual Machine; and

said second processor comprises a Java Card™ Virtual Machine.

25. The program storage device of claim 15 wherein

said first base is used by said first processor for performing arithmetic operations on at least one data type, said at least one data type having a size less than the size of said first base; and

said second base is used by said second processor for performing arithmetic operations on said at least one data type, said second base having a size equal to the size of said at least one data type.

26. The program storage device of claim 15 wherein

said first processor comprises a 32-bit processor; and

said second processor comprises a resource-constrained 16-bit processor.

27. The program storage device of claim 23 wherein

said first base is used by said first processor for performing arithmetic operations on at least one data type, said at least one data type having a size less than the size of said first base; and

said second base is used by said second processor for performing arithmetic operations on said at least one data type, said second base having a size greater than the size of said at least one data type.

28. An apparatus for arithmetic expression optimization, comprising:

at least one memory having program instructions; and

at least one processor configured to use the program instructions to:

receive a first instruction defined for a first processor having a first base, said first instruction including an operator and at least one operand;

convert said first instruction to a second instruction optimized for a second processor having a second base when said at least one operand does not carry potential overflow beyond said second base or when said operator is insensitive to overflow, said second base smaller than said first base; and

convert to a wider base a third instruction that is the source of potential overflow associated with said at least one operand when said at least one operand carries the potential for overflow beyond said second base and when said operator is sensitive to overflow, said third instruction having been previously optimized, said wider base larger than said second base and smaller or equal to said first base.

29. The apparatus of claim 28 wherein said at least one processor is further configured to use the program instructions to discard previous conversion results of said third instruction before converting to a wider base.

30. An apparatus for arithmetic expression optimization, comprising:

means for receiving a first instruction defined for a first processor having a first base, said first instruction including an operator and at least one operand;

means for converting said first instruction to a second instruction optimized for a second processor having a second base when said at least one operand does not carry potential overflow beyond said second base or when said operator is insensitive to overflow, said second base smaller than said first base; and

means for converting to a wider base a third instruction that is the source of potential overflow associated with said at least one operand when said at least one operand carries the potential for overflow beyond said second base and when said operator is sensitive to overflow, said third instruction having been previously optimized, said wider base larger than said second base and smaller or equal to said first base.

31. The apparatus of claim 30 wherein said means for converting to a wider base further comprises a means for discarding previous conversion results of said third instruction before said converting to a wider base.

32. The apparatus of claim 30, further comprising a means for linking each instruction to successor instructions in all control paths.

33. A method of using an application software program including arithmetic expression optimization of at least one instruction targeted to a processor having a first base, the method comprising:

receiving the software program on a processor; and

executing the sequence of instructions on the processor.

34. The method of claim 33, further including storing the at least one instruction on a resource-constrained device.

35. A smart card having a microcontroller embedded therein, the smart card comprising:

a virtual machine being executed by a microcontroller, the virtual machine executing a software application comprising of a plurality of previously optimized instructions, the virtual machine comprising means for receiving optimized instructions, the optimized instructions being previously optimized for execution on a resource-constrained device, means for executing said instructions.